remote sensing

MDPI

Article

A Grid Feature-Point Selection Method for Large-Scale Street View Image Retrieval Based on Deep Local Features

Tianyou Chu, Yumin Chen *, Liheng Huang, Zhiqiang Xu and Huangyuan Tan

School of Resource and Environment Science, Wuhan University, Wuhan 430079, China; chutianyou@whu.edu.cn (T.C.); 2014301110077@whu.edu.cn (L.H.); xuzq97@whu.edu.cn (Z.X.); tanhuangyuan@whu.edu.cn (H.T.)

* Correspondence: ymchen@whu.edu.cn

Received: 7 October 2020; Accepted: 1 December 2020; Published: 4 December 2020

Abstract: Street view image retrieval aims to estimate the image locations by querying the nearest neighbor images with the same scene from a large-scale reference dataset. Query images usually have no location information and are represented by features to search for similar results. The deep local features (DELF) method shows great performance in the landmark retrieval task, but the method extracts many features so that the feature file is too large to load into memory when training the features index. The memory size is limited, and removing the part of features simply causes a great retrieval precision loss. Therefore, this paper proposes a grid feature-point selection method (GFS) to reduce the number of feature points in each image and minimize the precision loss. Convolutional Neural Networks (CNNs) are constructed to extract dense features, and an attention module is embedded into the network to score features. GFS divides the image into a grid and selects features with local region high scores. Product quantization and an inverted index are used to index the image features to improve retrieval efficiency. The retrieval performance of the method is tested on a large-scale Hong Kong street view dataset, and the results show that the GFS reduces feature points by 32.27–77.09% compared with the raw feature. In addition, GFS has a 5.27–23.59% higher precision than other methods.

Keywords: street view; image retrieval; convolutional neural networks; geo-localization

1. Introduction

Street view images can be used to analyze and solve many problems, such as street-level vegetation estimates [1] and leaf area indexes [2] or urban land use mapping [3]. The above researches premise is the known location street view images. However, some valuable or interesting street view images have no or blurry location information. Some photos are taken by devices without GPS or taken in urban environments with multipath error [4]. Moreover, some pictures lose the location information during the propagation, such as upload and download. Thus, it is necessary to find the street view images location, and the problem is attracting increasing attention [5–9].

The content of street view images provides effective clues to help locate the shooting location, which can provide more contextual information about the scene. The location of street view images can be identified and matched by retrieving a large-scale street view dataset. Street view image retrieval extracts visual information as image features, and then the features are encoded to improve retrieval efficiency and speed on the large-scale dataset. The encoding method clusters the image features and numbers them so that the query features are searched in the small cluster during retrieval. Finally, the similarity is computed between the query and dataset features. At present, the image features used

for street view image retrieval mainly include (1) handcrafted local features where the algorithm is designed manually and (2) features extracted from Convolutional Neural Networks (CNNs).

Street view image retrieval methods based on handcrafted local image features such as SIFT [10], SURF [11], and Hesaff SIFT [12] are usually used in combination with descriptor aggregation methods such as bag-of-words (BoW) [13,14], Fisher vector (FV) [15] and vectors of locally aggregated descriptors (VLAD) [16]. However, it takes a long time to train the codebook to encode features, and it does not perform well on large-scale retrieval tasks. Some work [7,17] has improved the traditional handcrafted local feature retrieval method to adapt to street view or landmark retrieval. Ref. [18] proposed a method for detecting confusing features based on traditional local features such as trees and road signs in the street view. This method reduces redundant functions and the data quantity, but the retrieval precision does not improve significantly. The work in [19] improved retrieval performance by appropriately representing buildings with repetitive structures. However, the method has high computational complexity because it needs to build an undirected graph to distinguish similar features. Ref. [20] proposed a method for weighting SIFT features using geographic tags, and a Gaussian filter was used to return the best result; however, the premise is that the query image has GPS information.

Recently, other methods based on CNN image features have been proposed [21–24], which show better performance than handcrafted local features [25]. CNN-based features include MaxPool [26], SumPool [27], CROW [28], RMAC [29], and GEM [22]. In addition, aggregation methods such as BoW [30], VLAD [6], and Fisher vectors [31] are applied to the output of the convolutional layer. To extract distinguishing features, attention mechanisms are widely used [21,32,33]. Before image matching, PCA or whitening [34,35] are implemented, which can help improve the retrieval precision. The paper [6] proposed the back-propagated VLAD layer, which is inserted into the CNN architecture to generate image features and performs experiments on Google Street View datasets. Ref. [8] used the attention model and spatial pyramid pooling to generate image features. Ref. [24] proposed a feature extraction method that describes first and then detects. Dense local features are selected by calculating the high absolute and relative saliency. The method does not require training, but it is computationally expensive, and stable retrieval performance cannot be maintained when the number of feature points is reduced. Ref. [23] combined image retrieval, feature-point detection, and feature extraction into one framework. The image retrieval precision was high, but it still had a large number of feature points.

In addition, in large-scale street image retrieval systems, methods such as KD tree [36], hash function [37], or product quantization [38] are usually used to encode image features and generate indexes to solve the problem of low retrieval efficiency and speed. The index includes cluster center information, features encoding information, mapping information between images and features, etc. The query features search for the nearest feature through the index. However, they are often accompanied by a decrease in retrieval precision. Additionally, postprocessing techniques are used to improve retrieval performance, such as geometric verification [39], query expansion [32], and database-aside feature augmentation [40]. Geometry verification eliminates the images that have similar visual content but are different objects by random sample consensus (RANSAC) [41–43] or Hamming embedding [44] algorithms. Ref. [16] proposed a method based on Hamming embedding [17] to estimate the distinctiveness of features and remove insignificant features to improve the retrieval speed of the image retrieval system and reduce storage requirements, but only a small percentage of features can be removed.

The above methods of street view image retrieval have certain limitations. The method based on handcrafted local features does not need to generate a training dataset and clean up the noise data, but the time to train the codebook is long, the quantity of data is large, and it does not show practical retrieval performance. The method based on global features extracted from CNNs shows faster extraction and retrieval speed, and the quantity of data is small, but it cannot represent the local area of street view and is susceptible to the influence of interfering objects such as vehicles and pedestrians. Compared with global features, local features have better precision [45]. The method based on local features extracted from CNNs can represent the local area and show better retrieval

performance. However, the quantity of local feature data is too large, and it is difficult to apply to large-scale street view image retrieval.

Therefore, this paper proposes a grid feature-point selection method (GFS) based on deep local features. The method deletes low attention score feature points to reduce the feature file size and minimize the precision loss. The CNNs are constructed first to extract attentive and multiscale deep local features (DELF) that learn the weight of the object in the image by using the attention mechanisms during the training process and only require weakly-supervised classification. Then, distinctive features are selected by GFS to reduce memory usage so that local features can be applied to large-scale street view image retrieval. In this step, the image is divided into multiple grid regions, and features are mapped to the corresponding regions. According to the attention score, the features in each region are selected to remove redundant features. In addition, product quantization and the inverted index are used to improve retrieval efficiency. An ANN search is performed to retrieve query images. GFS compresses the number of feature points by selecting features with higher weights in the local region and shows higher accuracy compared with other methods.

2. Methodology

The GFS method includes the following: (1) data preprocessing, (2) street view image feature extraction based on DELF, (3) a grid feature-point selection method, (4) large-scale street view image retrieval based on product quantization, and (5) retrieval method comparison and evaluation. The workflow of the paper is shown in Figure 1.

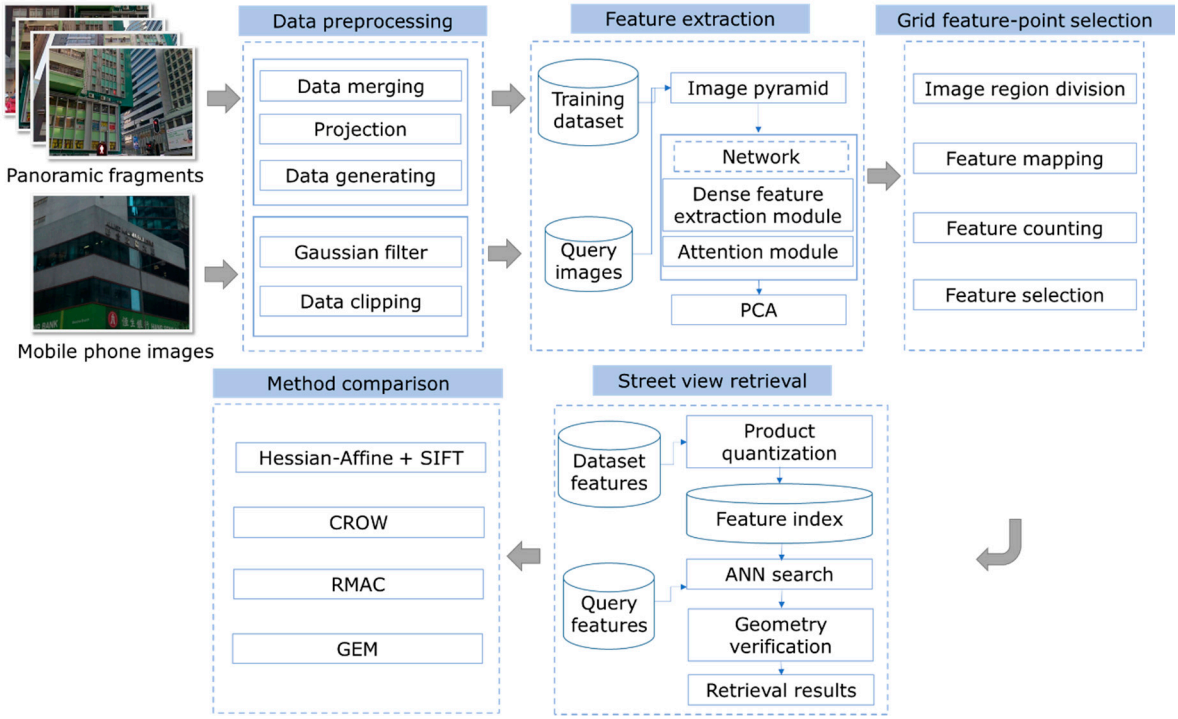


Figure 1. The street view image retrieval workflow.

2.1. Data Preprocessing

The geotagged perspective view images extracted from panoramas are used as the retrieval and training dataset, and images taken by a mobile phone are used as the query image. As shown in Figure 2, the original images of the dataset are the fragments of the equirectangular panoramic image collected by the camera-equipped on the vehicle. For this reason, multiple fragments are merged together into a complete panorama. The deformation at the center of the image is small, while the upper and lower sides are large. Considering that the deformation of the panoramic image is inconsistent

with the query image and the matching results are poor [7], the panoramic images are projected into the same perspective view as the query image so that the similarity between their features is improved. In more detail, the street view image preprocessing is as follows:

(1) Merge image fragments into a complete street view panorama. The equirectangular panorama is a single image with an aspect ratio of 2:1, so crop the black part of the panorama and keep the proper aspect ratio of the image.
(2) Set projection parameters. Parameters include FOV (field of view), pitch, and heading. The FOV determines the field of the projection, while the pitch and heading determine the location of the projection. The larger the FOV is, the larger the region is covered by the image. However, if the FOV is too large, the edges of the perspective image will be deformed. Thus, there is a trade-off between the image coverage area and the image distortion. The recommended FOV range is 40°–80°.
(3) Project panorama into perspective view image. Specifically, the projection process is divided into two steps. First, the equirectangular panorama is mapped to a sphere. Then, according to the projection parameters, an image of a certain range of the sphere is mapped to a plane to obtain a perspective view image.
(4) Generate training data. Every three neighboring panoramas are grouped into a class by querying the nearest neighbor images. The images with different orientations in the class are removed so that the image with the same scenes is left. Some images facing a certain scene in each class are retained.

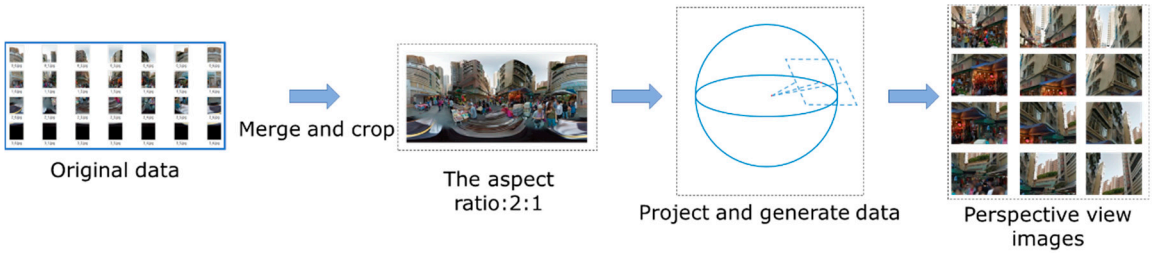


Figure 2. The street view panorama preprocessing flow.

The photos taken from the mobile phone have high resolution that slows down the process of feature extraction. In addition, the upper part of the photo usually has not a corresponding street view in the dataset or has a large deformation image, and the lower part of that is roads and pedestrians. Therefore, the photos are center cropped and resized to 640 × 480 size, and a Gaussian filter is finally performed to remove the noise.

2.2. Street View Image Feature Extraction Based on DELF

Attention-based multiscale deep local features are used to represent street view images. The attentive module can learn the targets with the same semantic information in the samples of each class and increase their feature weights, such as buildings and signs. DELF [21] is used to extract attention-based features. Image pyramids are constructed by resizing images and fed into CNNs to generate multiscale features that have the ability to deal with scale changes. A CNN convolutional layer is used to extract dense local features that are sorted according to the scores provided by the attention layer, and the features with lower scores are removed. The street view feature extraction network has two modules, which are trained separately, as follows:

(1) Dense feature extraction module. A fully convolutional neural network is used to extract the dense features. ResNet50 [46] is employed for training, and the output of block 4 is used for dense features. The module is trained in the first step. A total of 1500 classes of the Google Landmark dataset [21] and 500 classes of the San Francisco dataset [7] are used to train with 100 epochs,

and then the 2000 classes of the Hong Kong street view dataset are used to fine-tune 100 epochs. The cross-entropy loss function is adopted.

(2) Attention module. The attention score function $\propto (f_n; \theta)$ is constructed, θ represents the weight of each feature vector f_n, and $n(1, \ldots, N)$ represents the n^{th} feature vector. $f_n \in R^d$, d is the size of f_n, which depends on the dimensionality of the outputs of the convolution layer. The output f'_n is the weighted features, which is given by

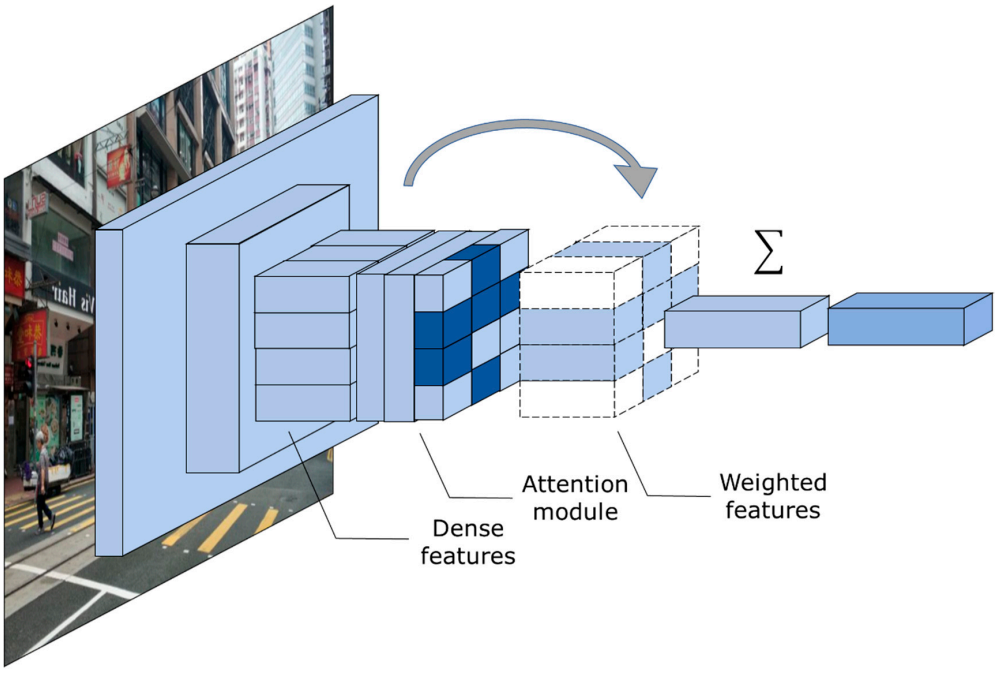
$$f'_n = \propto (f_n; \theta) \cdot f_n \tag{1}$$

As shown in Figure 3, two convolution layers with the softplus function are embedded into the dense feature extraction network in the second step. The subsequent pooling layer and fully connected layer are added to predict classes. The module is trained in this step singly, which means that the weights of dense feature extraction modules are frozen during the training process. The image pyramids generated from the Hong Kong street view dataset are used as a training dataset with 100 epochs. The cross-entropy loss function is adopted. The output of the fully connected layer y is the sum of weighted features, which is given by

$$y = W\left(\sum_n f'_n\right) \tag{2}$$

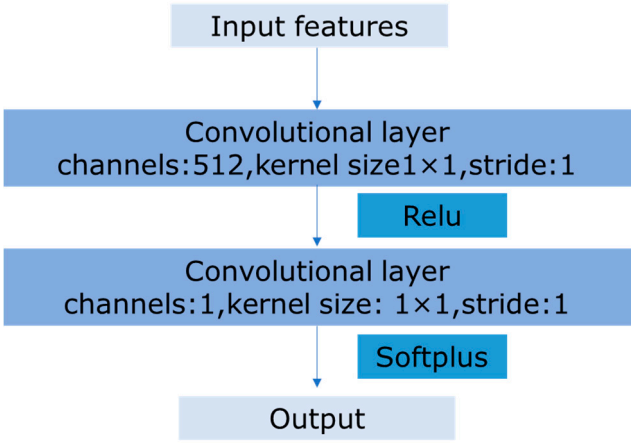
where $W \in R(M \times d)$ represents the weights of the final fully connected layer of the CNNs trained to predict M classes.

(3) Feature extraction. The image pyramids are fed into the network, and the network after the attention layer is deleted to extract attention-weighted features. Then, the features are selected according to the attention score. The top N features with the highest attention score for each image are selected. Finally, L2-normalize, PCA, and L2-normalize again are performed on each feature. In addition, the results also contain the position of the feature in the image, the attention score, and the feature scale.



(**a**) The street view image feature extraction architecture.

Figure 3. *Cont.*



(**b**) Attention module

Figure 3. Convolutional Neural Networks architecture and attention module.

2.3. A Grid Feature-Point Selection Method

Although the precision of local features is better than that of global features, an image requires a large number of local features for a description, which also leads to the need for more storage space. Due to a large number of images in a street view image retrieval system, the feature files take up considerable hard disk storage space. Table 1 shows the relationship between the number of feature points and the size of the feature file. It is an impossible task to load them directly into memory for image retrieval. Thus, product quantization is used to compress the image features for more efficient coding. It would be presented in Section 2.4. However, when performing product quantization, the original feature file also needs to be loaded into the GPU or memory for the training codebook. Although the CPU can be used to index features, the training speed is slow, while the execution speed on the GPU is dozens or even hundreds of times that of the CPU. However, graphics memory has limitations and hardly loads all feature files. Therefore, it is necessary to reduce the image feature file to an appropriate size, use as few distinctive features as possible to represent the images, and minimize the loss of precision. The effective feature selection method can reduce the size of the feature file and improve the speed of index construction.

Table 1. The relationship between the number of feature points and the size of the feature file.

No. of feature points	6650	12,499	34,736
Feature file size (GB)	6.1	11	38.2

To reduce memory consumption, the grid feature-point selection method (GFS) is proposed to reduce the features. As shown in Figure 4, the method is different from methods that simply select the first N features with high attention scores, but the image is divided into multiple regions uniformly by a grid and selects the first N features in each region. The method steps are as follows:

(1) The street view image of $H \times W$ size is divided into $I \times J$ regions, and the size of each region is $h \times w$. Each region contains corresponding image features. Features are located in the region of the image based on the receptive field of CNNs, which is calculated by the configuration of the convolution layer and the pooling layer. The center pixel coordinate of the receptive field is used for the position of the feature. In addition, the size of the receptive field is inversely proportional to the scale when performing multiscale feature extraction. The k^{th} feature of an image is T_k,

and its position in the image is (x_k, y_k). The feature is located in the (i, j) region, and $G_k(i_k, j_k)$ is the region number corresponding to the feature T_k:

$$\begin{array}{l} i_k = [x_k/w],\ i_k \in (1, I) \\ j_k = [y_k/h],\ j_k \in (1, J) \end{array} \tag{3}$$

(2) The number of feature points in each region $M_{i,j}$ is counted according to G_k.

(3) Finally, two strategies for selecting feature points are proposed.

a. GFS-N: The first strategy is to select by number, which means select the first N features with a high attention score in each region and all features in the grid are selected if the number of features in the grid is less than N, and $S_{i,j}$ is the feature number to be selected in each region (i, j):

$$S_{i,j} = Min(N, M_{i,j}) \tag{4}$$

b. GFS-P: The second strategy is to select by percentage, which means select the top n% features in each region. Actually, GFS-P equivalent to select high score feature points without a grid. Therefore, the method is proposed to compared with GFS-N to evaluate GFS performance.

$$S_{i,j} = n\% \times M_{i,j} \tag{5}$$

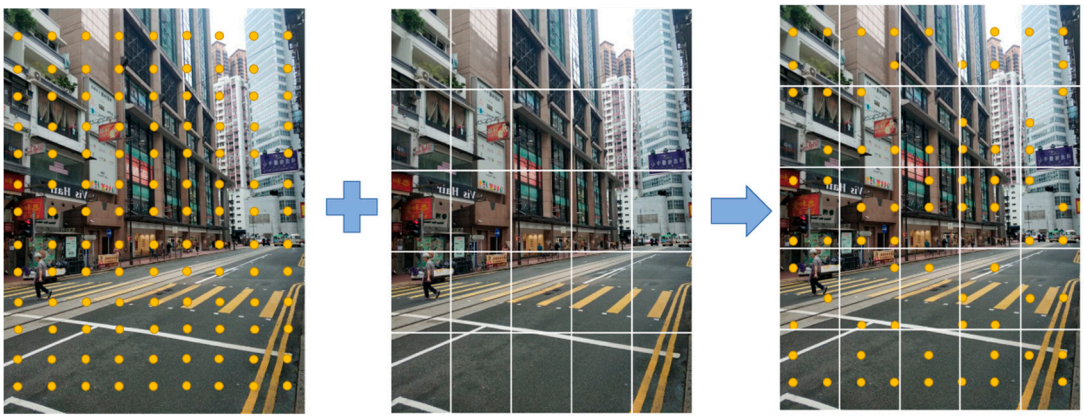


Figure 4. The grid feature-point selection method. Since a picture has hundreds of feature points, the figure is simplified. The yellow points represent the location of the feature points, and the white grid divides the image into multiple regions. After performing grid feature-point selection method (GFS), the number of feature points is reduced.

2.4. Large-Scale Street View Image Retrieval Based on Product Quantization

The street view image retrieval method searches the feature vector closest to the query vector from the features dataset, but it is not an effective method that traverses all features for a large-scale image retrieval system because it will take considerable time to calculate the distance among millions of vectors. Therefore, product quantization is deployed to construct an image index that improves the speed of retrieval and reduces memory requirements during retrieval.

Product quantization divides the N D-dimensional features into M parts first, and each part has N D/M-dimensional features. For example, N 128-dimensional features are divided into 4 parts, and each part has N 32-dimensional features. Next, each part is trained to a codebook separately. The size of the codebook is K. Then, each part of the feature generates an index value, which is the number of the nearest cluster center of this part. Finally, each feature is represented as an index value combined with the M parts index. The process is shown in Figure 5. To improve the precision of image retrieval,

the asymmetric distance calculation is performed, where the original features of the query image do not undergo the product quantization process. The distance of two image features is converted into the Euclidean distance of the indexes. The resulting image ID can be queried according to the image features through the inverted index file. The task is completed using Faiss [47], which is a library for similarity searching and clustering of vectors.

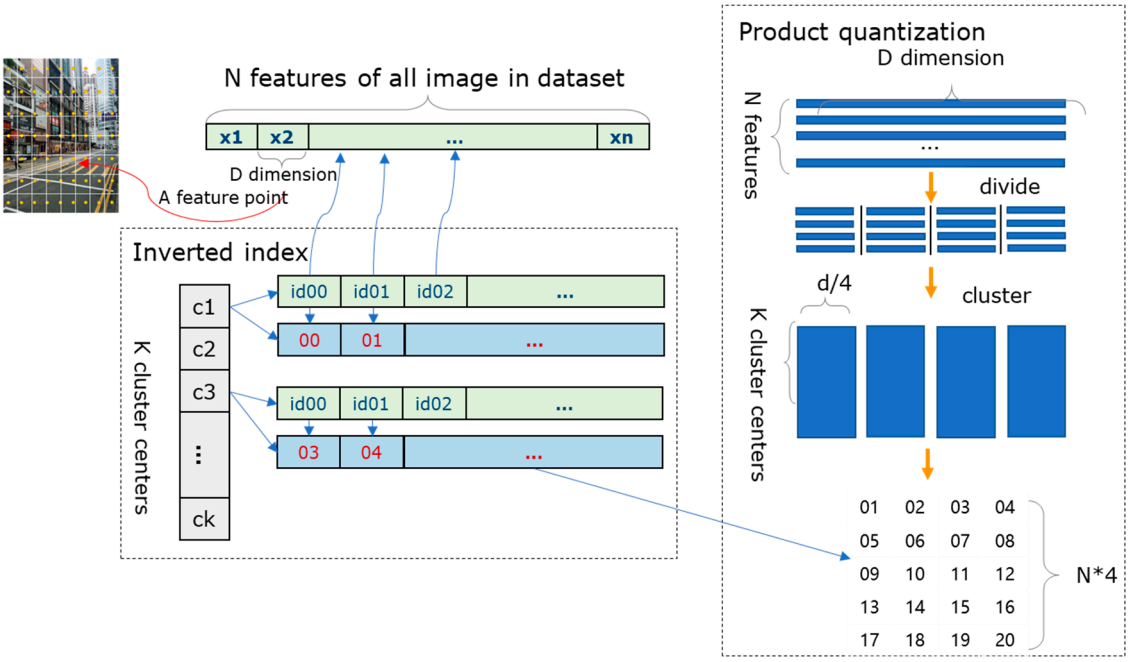


Figure 5. The process of the inverted index and product quantization.

An ANN (approximate nearest neighbor) search is performed for each local feature of the query image. As shown in Figure 6, all the retrieval results are summarized and correlated with the street view images in the dataset if there are k features that match the query image. The process is as follows:

(1) A query image Q contains n features $q_i (i \in 1 \ldots n)$, and an ANN search is performed on each feature q_i to obtain the most similar top m features $f_i (i \in 1 \ldots m)$ on the index of the feature database. Performing n queries will obtain $n * m$ features.

(2) Since each feature f_i corresponds to a street view image in the inverted index file, it is easy to count the number of similar features in each street view image and sort the results accordingly.

(3) Geometric verification is performed using RANSAC on retrieval results to exclude some distractor images that match the query image using ANN but are different from visual information. The retrieval results are sorted according to the number of inliers, and output.

2.5. Retrieval Method Comparison and Evaluation

The results of the method in this paper are compared with GEM [22], CROW [28], RMCA [29], and Hessian-Affine extractor + SIFT descriptor (Hesaff SIFT) [12] to evaluate the ability of street view image retrieval performance. The GEM, CROW, and RMCA are global features based on CNNs. Thus, the network of the dense feature extraction module with the same configuration in Section 2.2 is used, and dense features are pooled to extract the global features. The Hesaff SIFT method is a handcrafted local feature that has better performance than SIFT on large-scale retrieval tasks. Product quantization is used to generate indexes for retrieval features. In addition, all methods are tested on the Hong Kong street view dataset.

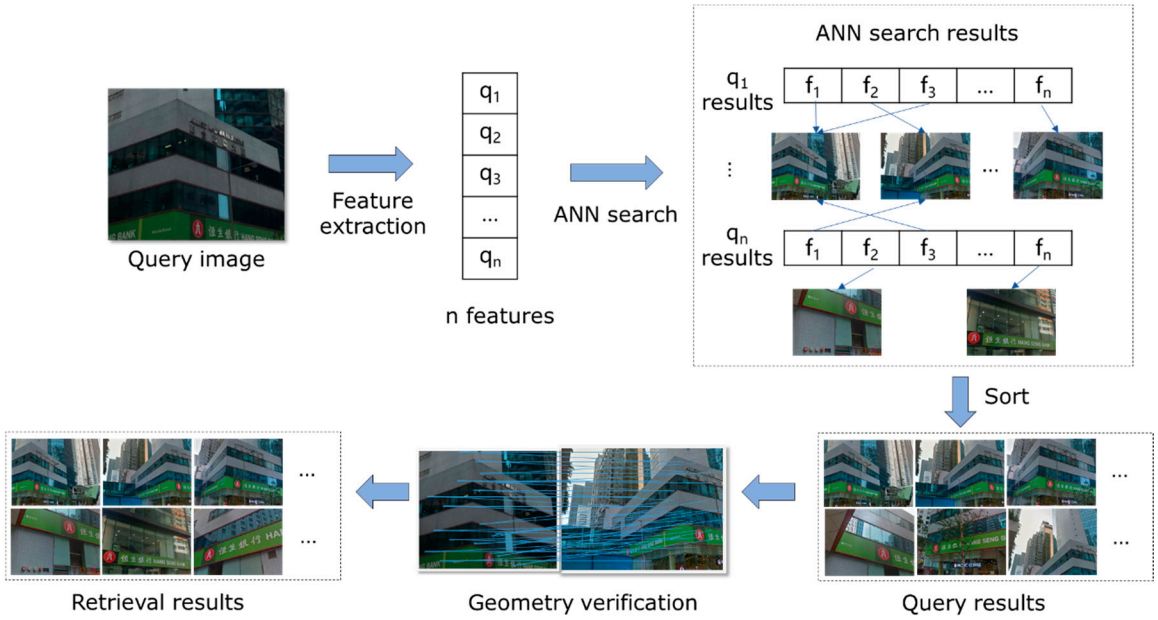


Figure 6. The approximate nearest neighbor (ANN) search workflow.

In this paper, the same visual information between the results and query image is considered. The evaluation measure P_v is employed, which is given by

$$P_v = \frac{\sum_{i=1}^{Q} q_i}{N}, \tag{6}$$

where Q is the number of query images, and if at least one image that has the same visual content is retrieved within the first N results for i^{th} query image, $q_i = 1$; otherwise, $q_i = 0$.

In addition, the distance between the retrieval results and the query image is also evaluated. The P_r in different radii of the query image is also counted, which is given by

$$\begin{aligned} P_r &= \frac{\sum_{i=1}^{Q} r_i}{N}, \\ r_i &= \begin{cases} 1 \; if \; \min(d_j) \le D \\ 0 \; if \; \min(d_j) > D \end{cases} \quad j \in (0, N), \end{aligned} \tag{7}$$

where d_j denotes the distance between the query image and j^{th} results; if at least one image that is in the range of query image radius (D) is retrieved within the first N results for the i^{th} query image, $r_i = 1$; otherwise, $r_i = 0$.

3. Experiments and Results

3.1. Study Area

The Causeway Bay and Wan Chai areas in the northern part of Hong Kong Island are selected as the study area, which is famous for a large number of skyscrapers with diverse building façades. The area is dense with roads, population, and buildings, and has a high rate of street view image collection. The buildings, especially residential buildings, have similar styles. Reflection glass curtainwalls, numerous vehicles, and pedestrians usually represent a considerable challenge for street view image retrieval.

The Hong Kong street view dataset in the study area contains 239,400 images (640 × 480) from 6650 panoramic classes collected in 2017 and 38 query images taken by mobile phones collected in 2019. The experimental data are within this range of (114.16267, 22.283887) to (114.18704, 22.273796), and the

regional area is 2.81 km^2. The dataset covers almost all major roads in the area. Each class is labeled with a GPS coordinate, and the distance between each class is approximately 10–12 m. The geographic distribution of these images is shown in Figure 7, and example images are presented in Figure 8. It is worth noting that the dataset images and the query images are not taken at the same time, which means that the billboards on the building facade may have been replaced or the buildings may have been renovated for other styles. There are obstacles in the image, such as vehicles and pedestrians. Both of them also create difficulties in the retrieval task.

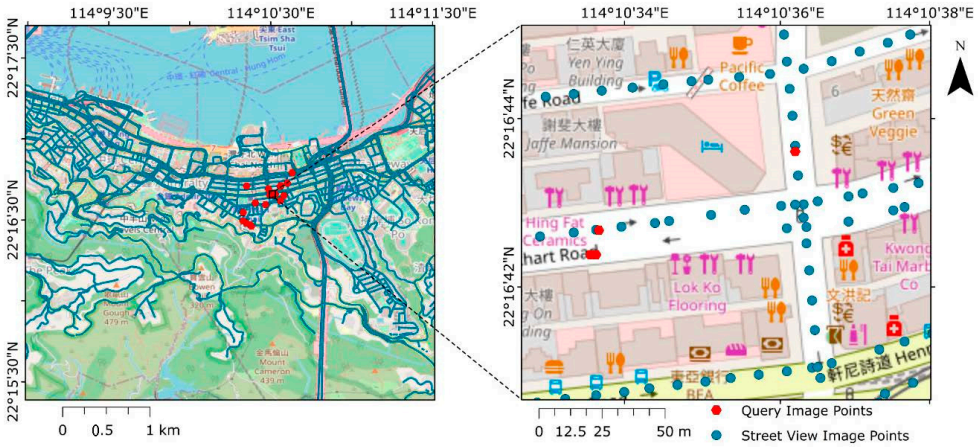
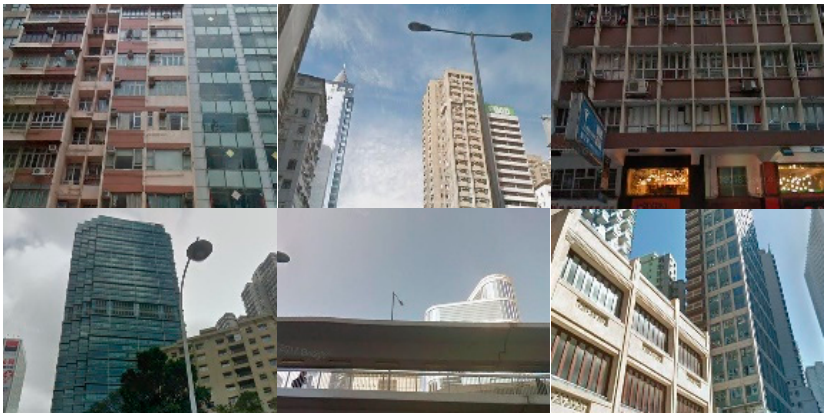


Figure 7. The distribution of street view images.

(**a**) Example query images



(**b**) Example dataset images

Figure 8. *Cont.*

(**c**) Example panoramas

Figure 8. Example street view images and query images.

3.2. Experiments Design

The experiments are completed on a computer equipped with an RTX 2080TI GPU, an INTEL I7-9700K CPU, and 32 GB RAM. The detailed implementation of the experiment is as follows:

(1) Data preprocessing. Projection parameters are set to crop the panoramic image and generate perspective images with approximately 20% overlap between the two adjacent images. The parameters set and other information are shown in Table 2. Each combination of parameters generates an image, and a panoramic image produces a total of 36 images.

(2) Street view image feature extraction based on DELF. As shown in Table 3, a configuration similar to [21] is adopted during the training process, and CNNs are built using PyTorch. The first 1000 local features are extracted based on the attention score.

Table 2. The street view dataset information.

FOV	50°
Pitch	[5°, 20°, 35°]
Heading	[0°, 30°, 60° , . . . , 330°]
Image size	640 * 480
Image quality	80% of the original panorama

Table 3. The network configuration for feature extraction.

Dense module training step	center crop (256 × 256), random crop (224 × 224)
Attention module training step	center crop (900 × 900), random crop (720 × 720)
Image pyramid scales	(0.25, 0.35, 0.5, 0.71, 1, 1.41, 2)
Batch size	16
Optimizer	SDG
Initial learning rate	0.008
PCA dimension	40

(3) Grid feature-point selection method: After experimenting with different sizes and numbers of grids, it was found that changes in the number and size of grids had no significant impact on the retrieval precision and the number of features. Therefore, as shown in Table 4, the grid is set to 8 × 10, and N (GFS-N) and n% (GFS-P) are set to different values to measure the impact on retrieval, where N = all refers to retaining all points in the grid region.

(4) Large-scale street view image retrieval based on product quantization. The index parameters are set through the index factory function in Faiss, which is 'OPQ10_40, IVF65536_HNSW32, PQ20'. Ten percent of features are used for training 65,536 centroids. After constructing the index, for each image feature of a query image, the 200 most similar feature points are retrieved, and the top 50 most similar street view images corresponding to these features are used for geometric verification. Finally, the results are output.

(5) Method comparison and evaluation. A brute-force search is performed on global features. Faiss is used to create an index of Hesaff SIFT, and the ANN search mentioned in Section 2.4 is performed for retrieval.

Table 4. GFS parameter settings.

Grid	8×10
N (GFS-N)	1, 2, 3, 6, 8, 10, 15, all
n% (GFS-P)	10%, 20%, 30%, . . . , 100%

3.3. Results

As shown in Table 5, GFS-N is evaluated on the Hong Kong street view dataset. Feature points refer to the average number of features per image after performing the method. The index size refers to the size of the image database index file generated from product quantization. The precision of the top1 retrieval results is presented. The precision of image retrieval increases as the value of N increases until it reaches the highest value when N = 8. After that, the precision of the model grows slowly. The size of the index file and the number of feature points are proportional to N.

The top 1 retrieval result of GFS-P is shown in Table 6. Since the number of feature points is proportional to the size of the index file, only the number of feature points is counted. The retrieval precision gradually increases with the proportion of feature selection. When it increased to 70%, the growth rate decreased. Compared with 100%, the precision lost 5.27%, and the number of feature points decreased by 22.02%. The percentage is positively related to the number of feature points.

As shown in Table 7, GFS (N = 8) outperforms other methods. The top 1 results are counted. The P_v of GFS is 5.27–23.59% higher than those of the other methods. Compared with other methods, the $P_{r=100}$, $P_{r=200}$, and $P_{r=500}$ of GPS increased by 13.16–26.32%, 10.89–26.32%, and 0–21.05%, respectively.

Table 5. Results of selecting features by number.

N	**1**	**2**	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**All**
No. of feature points	61	115	187	247	340	386	421	446	464	502
Index size (GB)	0.8	1.1	2	2.7	3.3	3.7	4.1	4.3	4.5	4.8
P_v	44.74	60.53	68.42	73.68	78.95	78.95	76.32	76.32	76.32	76.32

Table 6. Results of selecting features by percentage.

n	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
No. of feature points	71	109	156	202	251	297	347	396	445	502
P_v	52.63	57.89	57.89	60.53	68.42	68.42	71.05	68.42	76.32	76.32

Table 7. Results of the method comparison.

Method	**GFS**	**GEM**	**RMAC**	**CROW**	**Hesaff SIFT**
P_v	**78.95%**	65.79%	73.68%	55.26%	63.16%
$P_{r=100}$	**76.32%**	57.89%	63.16%	50.00%	63.16%
$P_{r=200}$	**84.21%**	68.42%	76.32%	57.89%	73.68%
$P_{r=500}$	**86.84%**	76.32%	65.79%	78.95%	**86.84%**

4. Discussion

4.1. The Evaluation of Compression Capability and Precision of GFS

GFS has the ability to effectively reduce feature volume and memory usage, which alleviates the problem that deep local features have too many feature points to load into memory in large-scale street

view image retrieval. Figure 9 shows the precision of GFS-N and GFS-P at different feature points. Overall, combined with Tables 5 and 6, it is shown that GFS-N is better than GFS-P in the range of 115 to 464 feature points that mean the performance of GFS is better than the method that selects high score feature points without a grid. In addition, according to the different application scenarios, GFS can reduce the number of feature points in the range of 32.27–77.09%. The feature points can be effectively reduced by GFS-N between 340 and 446 of the original points without loss precision performance. Compared to the results without any feature-point filtering (N = all), the number of feature points at the peak of precision (N = 8) is reduced by 32.27%, and precision increases by 2.63%. The number of feature points is reduced by 50.80% (N = 6), and the precision decreases by 2.64%. Further, the number of feature points is reduced by 77.09% (N = 2), and the precision decreases by 15.79%. GFS filters out the features with the smallest attention scores. Most of these features express interfering objects, which reduces the precision of the ANN search. The grid can select the high attention score feature points of the local region and keep richer feature points in the image for retrieval. In GFS-N, since N limits the upper limit of the number of feature points in each grid area, if the number of feature points in a grid area is less than N, the area will not be affected by N.

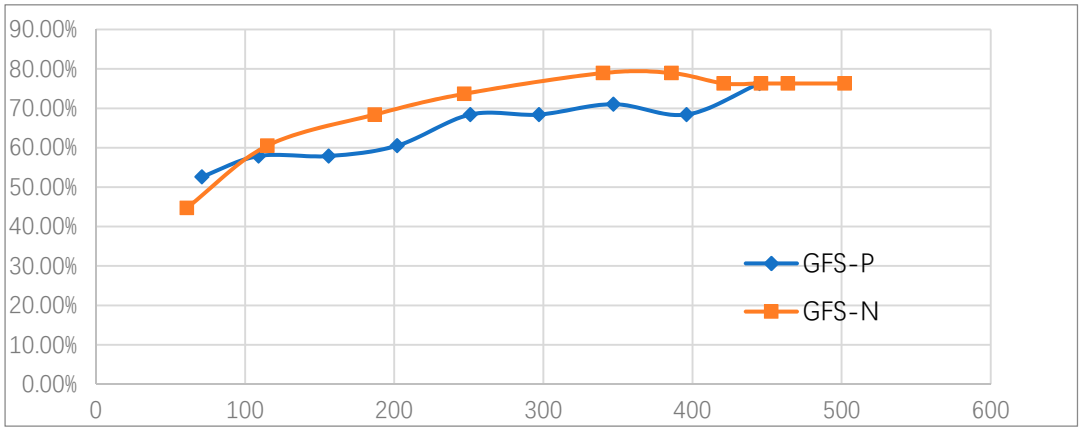


Figure 9. The precision of the two methods at different feature points.

4.2. Comparision with Other Methods

Compared with the GEM, RMAC, CROW, and Hesaff SIFT methods, GFS shows better precision. P_v of GFS (N = 8) increased by 13.16%, 5.27%, 23.69%, and 15.79%, respectively. Figure 10 shows a comparison of the results with other methods. In most cases, the retrieval system correctly retrieves the same scene as the query image. However, due to the shooting location, urban environment, and weather, the query image will have different content from the dataset, so it is necessary to extract the image features with the invariant angle or scale from the image. GFS has the ability to retrieve images with different angles but with the same semantic content, while other methods may retrieve results that are similar in content but are actually incorrect such as the sixth query image. There may be many reasons for GFS retrieval failure. The selected feature samples are not representative during the index training process, which causes the query features to search in the wrong cluster, such as the fourth and the fifth query images. As shown in Figure 11, although the recall of most methods in large-scale datasets is not good, GFS still has a high recall of correctly retrieved images. In addition to retrieving street scenes similar to the query image overall, the method can also extract features of a small area of the image, such as signs, billboards, and the building textures. The correct extraction of deep local features has a decisive influence on the retrieval results. This indicates that GFS has the ability to select representative features.

Figure 10. Comparison of the results with other methods. The green boxes are the correct result, while the red boxes are the incorrect result.

Figure 11. Comparison of the results with other methods. The green boxes are the correct result, while the red boxes are the incorrect result. Top1 refers to the image most similar to the query image, and so on.

4.3. Limitations and Future Enhancements

GFS is proposed to reduce the storage requirements and shows excellent performance. However, there are still some limitations. Although GFS can reduce the number of feature points and index file size, a regular rectangular grid cannot filter features of irregular objects. Furthermore, since there are a large number of uncertain interference factors such as vehicles and pedestrians in the street view image, it is difficult for the regular grid to effectively filter the features expressing the interferences. Therefore, the interference information in the image can be purposefully filtered through other methods, such as semantic segmentation [48,49] or object detection [50,51], to reduce their contribution in similarity calculation. Moreover, because of the lack of research on feature compression currently, GFS will be compared with other future research methods.

In addition, experiments are only performed on a partial area of the city, and the query data coverage is relatively small. Consideration should be given to different urban environments, such as snow or night. Further experiments will also be conducted in a more varied environment.

Finally, because the shooting location of the street view image is usually different from the query image, there are usually some errors using only the visual information of the image to retrieve the location. It is possible to combine with 3D reconstruction [52] to improve the precision of street view localization.

5. Conclusions

This paper proposes a grid feature-point selection method (GFS) suitable for large-scale street view image retrieval based on deep local features. Attention-based multiscale features are extracted to represent street view images. The grid is used to divide the image into several rectangular regions and selects a certain number of features in each region to reduce the number of feature points. Product quantization is performed to construct an index of features and speed up image retrieval. The Hong Kong street view dataset and mobile phone photos are used in experiments. The results show that the GFS can select representative local features and reduce the number of feature points by 32.27–77.09% compared with raw feature points. In addition, GFS outperforms other methods in retrieval precision. Future exploration will also focus on selecting more representative features and improving the robustness of retrieval in a variety of urban environments.

Author Contributions: Methodology, T.C., Y.C., and L.H.; software, T.C. and L.H.; validation, Z.X. and H.T.; writing—original draft preparation, T.C. and Y.C.; visualization, Z.X. and H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key S&T Special Projects of China [Grant No. 2017YFB0503704].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lauko, I.G.; Honts, A.; Beihoff, J.; Rupprecht, S. Local color and morphological image feature based vegetation identification and its application to human environment street view vegetation mapping, or how green is our county? *Geo Spat. Inf. Sci.* **2020**, *23*, 222–236. [CrossRef]
2. Richards, D.; Wang, J.W. Fusing street level photographs and satellite remote sensing to map leaf area index. *Ecol. Indic.* **2020**, *115*, 8. [CrossRef]
3. Chang, S.Z.; Wang, Z.M.; Mao, D.H.; Guan, K.H.; Jia, M.M.; Chen, C.Q. Mapping the Essential Urban Land Use in Changchun by Applying Random Forest and Multi-Source Geospatial Data. *Remote Sens.* **2020**, *12*, 2488. [CrossRef]
4. Chen, Q.X.; Ding, D.D.; Wang, X.; Liu, A.X.; Zhao, J.F. An efficient urban localization method based on speed humps. *Sust. Comput.* **2019**, *24*, 9. [CrossRef]
5. Ozaki, K.; Yokoo, S. Large-scale Landmark Retrieval/Recognition under a Noisy and Diverse Dataset. *arXiv* **2019**, arXiv:1906.04087.

6. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5297–5307. [CrossRef]
7. Chen, D.M.; Baatz, G.; Köser, K.; Tsai, S.S.; Vedantham, R.; Pylvänäinen, T.; Roimela, K.; Chen, X.; Bach, J.; Pollefeys, M. City-scale landmark identification on mobile devices. In Proceedings of the CVPR 2011, Providence, RI, USA, 20–25 June 2011; pp. 737–744.
8. Zhu, Y.Y.; Wang, J.; Xie, L.X.; Zheng, L. Attention-based Pyramid Aggregation Network for Visual Place Recognition. In Proceedings of the 26th ACM international conference on Multimedia, Seoul, Korea, 22–26 October 2018; pp. 99–107.
9. Weng, L.; Gouet-Brunet, V.; Soheilian, B. Semantic signatures for large-scale visual localization. *Multimed. Tools Appl.* **2020**. [CrossRef]
10. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
11. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
12. Perd'och, M.; Chum, O.; Matas, J. Efficient Representation of Local Geometry for Large Scale Object Retrieval. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; Volume 1–4, pp. 9–16.
13. Arulmozhi, P.; Abirami, M. Generation of Visual Patterns from BoVW for Image Retrieval using modified Similarity Score Fusion. *Adv. Electr. Comput. Eng.* **2020**, *20*, 101–112. [CrossRef]
14. Zhang, L.J. Feature mining simulation of video image information in multimedia learning environment based on BOW algorithm. *J. Supercomput.* **2020**, *76*, 6561–6578. [CrossRef]
15. Sukhia, K.N.; Riaz, M.M.; Ghafoor, A.; Ali, S.S. Content-based remote sensing image retrieval using multi-scale local ternary pattern. *Digit. Signal Process.* **2020**, *104*, 9. [CrossRef]
16. Liu, H.; Zhao, Q.J.; Mbelwa, J.T.; Tang, S.; Zhang, J.W. Weighted two-step aggregated VLAD for image retrieval. *Vis. Comput.* **2019**, *35*, 1783–1795. [CrossRef]
17. Torii, A.; Arandjelovic, R.; Sivic, J.; Okutomi, M.; Pajdla, T. 24/7 Place Recognition by View Synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 257–271. [CrossRef] [PubMed]
18. Knopp, J.; Sivic, J.; Pajdla, T. Avoiding Confusing Features in Place Recognition. In *Computer Vision-Eccv 2010, Pt I*; Daniilidis, K., Maragos, P., Paragios, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6311, pp. 748–761.
19. Torii, A.; Sivic, J.; Okutomi, M.; Pajdla, T. Visual Place Recognition with Repetitive Structures. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 2346–2359. [CrossRef] [PubMed]
20. Zamir, A.R.; Shah, M. Accurate Image Localization Based on Google Maps Street View. In *Computer Vision-Eccv 2010, Pt Iv*; Daniilidis, K., Maragos, P., Paragios, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6314, pp. 255–268.
21. Noh, H.; Araujo, A.; Sim, J.; Weyand, T.; Han, B. Large-Scale Image Retrieval with Attentive Deep Local Features. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3476–3485. [CrossRef]
22. Radenović, F.; Tolias, G.; Chum, O. Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1655–1668. [CrossRef] [PubMed]
23. Yang, T.-Y.; Nguyen, D.-K.; Heijnen, H.; Balntas, V. Ur2kid: Unifying retrieval, keypoint detection, and keypoint description without local correspondence supervision. *arXiv* **2020**, arXiv:2001.07252.
24. Tian, Y.; Balntas, V.; Ng, T.; Barroso-Laguna, A.; Demiris, Y.; Mikolajczyk, K. D2D: Keypoint Extraction with Describe to Detect Approach. *arXiv* **2020**, arXiv:2005.13605.
25. Zheng, L.; Yang, Y.; Tian, Q. SIFT meets CNN: A decade survey of instance retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1224–1244. [CrossRef]
26. Razavian, A.S.; Sullivan, J.; Carlsson, S.; Maki, A. Visual instance retrieval with deep convolutional networks. *ITE Trans. Media Technol. Appl.* **2016**, *4*, 251–258. [CrossRef]
27. Babenko, A.; Lempitsky, V. Aggregating Deep Convolutional Features for Image Retrieval. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Las Condes, Chile, 11–18 December 2015; pp. 1269–1277. [CrossRef]

28. Kalantidis, Y.; Mellina, C.; Osindero, S. Cross-dimensional weighting for aggregated deep convolutional features. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 685–701.
29. Tolias, G.; Sicre, R.; Jégou, H. Particular Object Retrieval with Integral Max-Pooling of CNN Activations. *arXiv* **2015**, arXiv:1511.05879.
30. Liu, X.B.; Zhang, S.L.; Huang, T.J.; Tian, Q. E(2)BoWs: An end-to-end Bag-of-Words model via deep convolutional neural network for image retrieval. *Neurocomputing* **2020**, *395*, 188–198. [CrossRef]
31. Ma, L.; Jiang, W.H.; Jie, Z.Q.; Wang, X. Bidirectional image-sentence retrieval by local and global deep matching. *Neurocomputing* **2019**, *345*, 36–44. [CrossRef]
32. Imbriaco, R.; Sebastian, C.; Bondarev, E.; de With, P.H.N. Aggregated Deep Local Features for Remote Sensing Image Retrieval. *Remote Sens.* **2019**, *11*, 493. [CrossRef]
33. Xiong, W.; Lv, Y.F.; Cui, Y.Q.; Zhang, X.H.; Gu, X.Q. A Discriminative Feature Learning Approach for Remote Sensing Image Retrieval. *Remote Sens.* **2019**, *11*, 281. [CrossRef]
34. Morere, O.; Lin, J.; Veillard, A.; Duan, L.-Y.; Chandrasekhar, V.; Poggio, T. Nested invariance pooling and RBM hashing for image instance retrieval. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, Bucharest, Romania, 6 June 2017; pp. 260–268.
35. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features off-the-shelf: An Astounding Baseline for Recognition. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 24–27 June 2014; pp. 512–519. [CrossRef]
36. Zhang, F.Q.; Gao, Y.H.; Xu, L.Q. An adaptive image feature matching method using mixed Vocabulary-KD tree. *Multimed. Tools Appl.* **2020**, *79*, 16421–16439. [CrossRef]
37. Shan, X.; Liu, P.; Gou, G.; Zhou, Q.; Wang, Z. Deep Hash Remote Sensing Image Retrieval with Hard Probability Sampling. *Remote Sens.* **2020**, *12*, 2789. [CrossRef]
38. Yang, J.C.; Chen, B.; Xia, S.T. Mean-removed product quantization for large-scale image retrieval. *Neurocomputing* **2020**, *406*, 77–88. [CrossRef]
39. Sivic, J.; Zisserman, A. Video Google: A Text Retrieval Approach to Object Matching in Videos. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; Volume 2, p. 1470.
40. Arandjelovic, R.; Zisserman, A. Three things everyone should know to improve object retrieval. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2911–2918.
41. Tran, N.T.; Tan, D.K.L.; Doan, A.D.; Do, T.T.; Bui, T.A.; Tan, M.X.; Cheung, N.M. On-Device Scalable Image-Based Localization via Prioritized Cascade Search and Fast One-Many RANSAC. *IEEE Trans. Image Process.* **2019**, *28*, 1675–1690. [CrossRef]
42. Li, X.Q.; Yang, J.S.; Ma, J.W. Large Scale Category-Structured Image Retrieval for Object Identification Through Supervised Learning of CNN and SURF-Based Matching. *IEEE Access* **2020**, *8*, 57796–57809. [CrossRef]
43. Zhan, Z.Q.; Zhou, G.F.; Yang, X. A Method of Hierarchical Image Retrieval for Real-Time Photogrammetry Based on Multiple Features. *IEEE Access* **2020**, *8*, 21524–21533. [CrossRef]
44. Yang, J.F.; Liang, J.; Shen, H.; Wang, K.; Rosin, P.L.; Yang, M.H. Dynamic Match Kernel with Deep Convolutional Features for Image Retrieval. *IEEE Trans. Image Process.* **2018**, *27*, 5288–5302. [CrossRef]
45. Cao, B.; Araujo, A.; Sim, J. Unifying Deep Local and Global Features for Image Search. *arXiv* **2020**, arXiv:2001.05027.
46. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
47. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with GPUs. *arXiv* **2017**, arXiv:1702.08734. [CrossRef]
48. Lin, C.Y.; Chiu, Y.C.; Ng, H.F.; Shih, T.K.; Lin, K.H. Global-and-Local Context Network for Semantic Segmentation of Street View Images. *Sensors* **2020**, *20*, 2907. [CrossRef] [PubMed]
49. Hao, S.J.; Zhou, Y.; Guo, Y.R. A Brief Survey on Semantic Segmentation with Deep Learning. *Neurocomputing* **2020**, *406*, 302–321. [CrossRef]
50. Xie, Q.; Li, D.W.; Yu, Z.H.; Zhou, J.; Wang, J. Detecting Trees in Street Images via Deep Learning with Attention Module. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 5395–5406. [CrossRef]

51. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
52. Doulamis, A.; Voulodimos, A.; Protopapadakis, E.; Doulamis, N.; Makantasis, K. Automatic 3D Modeling and Reconstruction of Cultural Heritage Sites from Twitter Images. *Sustainability* **2020**, *12*, 4223. [CrossRef]